

# Stabilizer Quantum Codes: A Unified View based on Forney-style Factor Graphs

Pascal O. Vontobel  
Hewlett-Packard Laboratories  
Palo Alto, CA 94304, USA  
Email: pascal.vontobel@ieee.org

**Abstract**—Quantum error-correction codes (QECCs) are a vital ingredient of quantum computation and communication systems. In that context it is highly desirable to design QECCs that can be represented by graphical models which possess a structure that enables efficient and close-to-optimal iterative decoding.

In this paper we focus on stabilizer QECCs, a class of QECCs whose construction is rendered non-trivial by the fact that the stabilizer label code, a code that is associated with a stabilizer QECC, has to satisfy a certain self-orthogonality condition. In order to design graphical models of stabilizer label codes that satisfy this condition, we extend a duality result for Forney-style factor graphs (FFGs) to the stabilizer label code framework. This allows us to formulate a simple FFG design rule for constructing stabilizer label codes, a design rule that unifies several earlier stabilizer label code constructions.

## I. INTRODUCTION

Graphical models have played a very important role in the recent history of error-correction coding (ECC) schemes for conventional channel and storage setups [1]. In particular, some of the most powerful ECC schemes known today, like message-passing iterative (MPI) decoding of low-density parity-check (LDPC) and turbo codes, can be represented by graphical models. It is therefore highly desirable to extend the design and analysis lessons that have been learned from these ECC systems to quantum error-correction code (QECC) systems, in particular to stabilizer QECC systems.

For background material and a history of stabilizer QECCs in particular, and quantum information processing (QIP) in general, we refer to the excellent textbook by Nielsen and Chuang [2]. Alternatively, one can consult some early papers on stabilizer QECCs, e.g. [3], [4], or more recent accounts, e.g. [5], [6], [7]. The aim of the present paper is to introduce a Forney-style factor graph (FFG) framework that allows one to construct FFGs that represent interesting classes of stabilizer QECCs, more precisely, that represent interesting classes of stabilizer label codes and normalizer label codes. Anyone familiar with the basics of stabilizer QECCs can then easily formulate the corresponding stabilizer QECCs. (Note that due to space constraints this paper does not motivate stabilizer QECCs and does not define them. However, the paper does not use any QIP jargon and should therefore be accessible to anyone familiar with the basics of coding theory.)

This paper is structured as follows. In Section II we introduce the basics on FFGs and in Section III we extend a well-known duality result for FFGs. In Section IV we then show

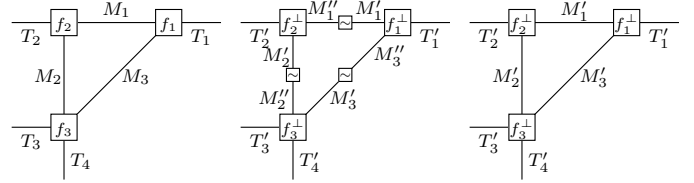


Fig. 1. Left: A simple FFG. Middle: Dual of the FFG on the left. Right: Dual of the FFG on the left if alphabet groups have characteristic 2.

how this duality result can be used to construct stabilizer and normalizer label codes. Afterwards, in Section V we discuss several examples of such codes, in particular we show that our FFG framework unifies earlier proposed code constructions. We conclude by briefly commenting on message-passing iterative decoding and linear programming decoding in Section VI. Because of space limitations we decided to formulate many of the concepts and results in terms of examples; most of them can be suitably generalized.

Our notation is quite standard. In particular, the field of real numbers will be denoted by  $\mathbb{R}$  and the ring of integers modulo  $p$  by  $\mathbb{Z}_p$ . (If  $p$  is a prime, then  $\mathbb{Z}_p$  is a field.) The Galois field  $\mathbb{F}_4$  will be based on the set  $\{0, 1, \omega, \omega^2\}$ , where  $\omega$  satisfies  $\omega^2 = \omega + 1$  (and therefore also  $\omega^3 = 1$ ), and where conjugation is defined by  $\bar{x} = x^2$ . Moreover, for any statement  $S$  we will use Iverson's convention which says that  $[S] = 1$  if  $S$  is true and  $[S] = 0$  otherwise.

## II. FORNEY-STYLE FACTOR GRAPHS (FFGs)

FFGs [8], [9], [1], also known as normal factor graphs, are graphs that represent multivariate functions. For example, let  $T = 4$  and  $M = 3$ , let  $T_i, i = 1, \dots, T$  and  $M_i, i = 1, \dots, M$  be some arbitrary alphabets, and consider the function  $f : \prod_{i=1}^T T_i \times \prod_{i=1}^M M_i \rightarrow \mathbb{R}$  that represents the mapping

$$(t, m) \mapsto f_1(t_1, m_1, m_3) f_2(t_2, m_1, m_2) f_3(t_3, t_4, m_2, m_3).$$

Here,  $f$  is called the global function and is the product of  $F$  functions  $f_i, i = 1, \dots, F$  (here  $F = 3$ ), which are called local functions. Whereas the argument set of the function  $f$  encompasses  $t_1, t_2, t_3, t_4, m_1, m_2$ , and  $m_3$ , the function  $f_1$  has only  $t_1, m_1$ , and  $m_3$  as arguments,  $f_2$  has only  $t_2, m_1$ , and  $m_2$  as arguments, and  $f_3$  has only  $t_3, t_4, m_2$ , and  $m_3$  as arguments. Graphically, we represent this function decomposition as follows, cf. Fig. 1 (left):

- For each local function we draw a function node (vertex).
- For each variable we draw an half-edge or an edge.
- If a variable appears as an argument in only one local function then we draw an half-edge that is connected to that local function. If a variable appears as an argument in two local functions then we draw an edge that connects these two local functions.<sup>1</sup>

In our example the variables that are associated with half-edges are labeled  $T_i$ ,  $i = 1, \dots, T$ , whereas the variables that are associated with edges are labeled  $M_i$ ,  $i = 1, \dots, M$ . This distinction of variable labels will be very helpful later on when we will dualize the global function.

Interesting are global functions where the local function argument sets are strict subsets of the global function argument set: the fewer arguments appear in the local functions, the sparser the corresponding FFG will be.

Consider again the FFG in Figure 1 (left) and let the local functions represent indicator functions, i.e.,

$$\begin{aligned} f_1(t_1, m_1, m_3) &\triangleq [(t_1, m_1, m_3) \in \mathcal{C}_1], \\ f_2(t_2, m_1, m_2) &\triangleq [(t_2, m_1, m_2) \in \mathcal{C}_2], \\ f_3(t_3, t_4, m_2, m_3) &\triangleq [(t_3, t_4, m_2, m_3) \in \mathcal{C}_3], \end{aligned}$$

for some codes  $\mathcal{C}_1 \subseteq \mathcal{T}_1 \times \mathcal{M}_1 \times \mathcal{M}_3$ ,  $\mathcal{C}_2 \subseteq \mathcal{T}_2 \times \mathcal{M}_1 \times \mathcal{M}_2$ , and  $\mathcal{C}_3 \subseteq \mathcal{T}_3 \times \mathcal{T}_4 \times \mathcal{M}_2 \times \mathcal{M}_3$ . The restriction of the resulting global function to the variables  $\mathbf{t}$ , i.e., to the variables that are associated with the half-edges, yields the function  $[\mathbf{t} \in \mathcal{C}]$  with

$$\mathcal{C} = \left\{ \mathbf{t} \in \prod_{i=1}^T \mathcal{T}_i \mid \begin{array}{l} \text{there exists an } \mathbf{m} \\ \text{such that } f(\mathbf{t}, \mathbf{m}) = 1 \end{array} \right\}.$$

Clearly, if the sets  $\mathcal{T}_i$ ,  $i = 1, \dots, T$ , and  $\mathcal{M}_i$ ,  $i = 1, \dots, M$ , are groups and the codes  $\mathcal{C}_i$ ,  $i = 1, \dots, F$  are group codes then  $\mathcal{C}$  is a group code.<sup>2</sup>

### III. DUALIZING FFGs

In the context of stabilizer QECCs, dual codes (under the symplectic inner product) play a very important role. The aim of this section is to start with an FFG that represents the indicator function of some code and to construct an FFG that represents the indicator function of the dual of that code. To that end we will heavily use insights from [8, Section VII] on Pontryagin duality theory in the context of FFGs, notably one of relatively few results that hold for graphical models *without* and *with* cycles.

<sup>1</sup>Global functions that contain variables that appear in more than two local functions can always be replaced by essentially equivalent global functions where all variables appear as an argument in at most two local functions. E.g.,  $f(x_1, x_2, x_3, x_4) = f_1(x_1, x_2) \cdot f_2(x_1, x_3) \cdot f_3(x_1, x_4)$  can be replaced by the essentially equivalent  $f'(x'_1, x''_1, x'''_1, x_2, x_3, x_4) = f_1(x'_1, x_2) \cdot f_2(x'_1, x_3) \cdot f_3(x'_1, x_4) \cdot [x'_1 = x_2 = x_3]$ . With this,  $f$  being “essentially equivalent” to  $f'$  means that whenever  $f(x'_1, x''_1, x'''_1, x_2, x_3, x_4)$  is nonzero then  $f(x_1, x_2, x_3, x_4) = f(x'_1, x''_1, x'''_1, x_2, x_3, x_4)$  with  $x_1 = x'_1 = x''_1 = x'''_1$ .

<sup>2</sup>A code is a group code if it is a subgroup of the direct product of the symbol alphabet groups. Note that a group code can be defined as the span of a list of suitably chosen vectors. Considering the group operation as “addition”, group codes can also be seen to be additive codes, i.e., codes that are closed under addition.

For the rest of the paper, we make the following definitions.

**Definition 1** Let  $p$  be some prime.

For  $i = 1, \dots, T$ :

- We define  $\mathcal{T}_i$  to be the group  $\mathbb{Z}_p^2$  with vector addition modulo  $p$  and we denote elements of  $\mathcal{T}_i$  by  $t_i = (t_{X,i}, t_{Z,i})$ .
- We let  $\mathcal{T}'_i$  be the character group of  $\mathcal{T}_i$ . Because  $\mathcal{T}'_i$  turns out to be isomorphic to  $\mathcal{T}_i$ , we identify  $\mathcal{T}'_i$  with  $\mathcal{T}_i$ . Elements of  $\mathcal{T}'_i$  will be denoted by  $t'_i = (t'_{X,i}, t'_{Z,i})$ .
- We define the inner product  $\langle t_i, t'_i \rangle : \mathcal{T}_i \times \mathcal{T}'_i \rightarrow \mathbb{Z}_p$  to be the symplectic inner product, i.e.,

$$\langle t_i, t'_i \rangle \triangleq \begin{bmatrix} t_{X,i} \\ t_{Z,i} \end{bmatrix}^T \cdot \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} t'_{X,i} \\ t'_{Z,i} \end{bmatrix} = t_{X,i} t'_{Z,i} + t_{Z,i} t'_{X,i},$$

where  $^T$  represents vector transposition and where addition and multiplication are modulo  $p$ . (Note that we are using angular brackets to denote inner products. This is in contrast to [8] that used angular brackets to denote pairings, which are exponential functions of inner products.)

For  $i = 1, \dots, M$ :

- We let  $\mu_i$  be some positive integer, we define  $\mathcal{M}_i$  to be the group  $(\mathbb{Z}_p^2)^{\mu_i}$  with vector addition modulo  $p$ , and we denote elements of  $\mathcal{M}_i$  by  $m_i = ((m_{X,i,1}, m_{Z,i,1}), \dots, (m_{X,i,\mu_i}, m_{Z,i,\mu_i}))$ .
- We let  $\mathcal{M}'_i$  denote the character group of  $\mathcal{M}_i$ . Again, because  $\mathcal{M}'_i$  turns out to be isomorphic to  $\mathcal{M}_i$ , we identify  $\mathcal{M}'_i$  with  $\mathcal{M}_i$ . Elements of  $\mathcal{M}'_i$  will be denoted by  $m'_i = ((m'_{X,i,1}, m'_{Z,i,1}), \dots, (m'_{X,i,\mu_i}, m'_{Z,i,\mu_i}))$ .
- We define the inner product  $\langle m_i, m'_i \rangle$  to be the symplectic inner product, i.e.,

$$\begin{aligned} \langle m_i, m'_i \rangle &\triangleq \begin{bmatrix} m_{X,i,1} \\ m_{Z,i,1} \\ \vdots \\ m_{X,i,\mu_i} \\ m_{Z,i,\mu_i} \end{bmatrix}^T \cdot \begin{bmatrix} 0 & 1 & & & \\ & 1 & 0 & & \\ & & \ddots & & \\ & & & 0 & 1 \\ & & & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} m'_{X,i,1} \\ m'_{Z,i,1} \\ \vdots \\ m'_{X,i,\mu_i} \\ m'_{Z,i,\mu_i} \end{bmatrix} \\ &= \sum_{h=1}^{\mu_i} (m_{X,i,h} m'_{Z,i,h} + m_{Z,i,h} m'_{X,i,h}). \end{aligned}$$

The above inner products induce inner products on vectors, e.g.,  $\langle \mathbf{t}, \mathbf{t}' \rangle : \prod_{i=1}^T \mathcal{T}_i \times \prod_{i=1}^T \mathcal{T}'_i \rightarrow \mathbb{Z}_p$  is the inner product defined by  $\langle \mathbf{t}, \mathbf{t}' \rangle \triangleq \sum_{i=1}^T \langle t_i, t'_i \rangle$ , etc..

Moreover, all codes are assumed to be group codes.  $\square$

In the following, because of the natural isomorphism of the groups  $(\mathbb{Z}_p^2)^n$  and  $(\mathbb{Z}_p^n)^2$ , a vector  $\mathbf{t} \in (\mathbb{Z}_p^2)^n$  will not only be written as

$$\mathbf{t} = ((t_{X,1}, t_{Z,1}), \dots, (t_{X,n}, t_{Z,n}))$$

but also as

$$\begin{aligned} \mathbf{t} &= (\mathbf{t}_X, \mathbf{t}_Z) \quad \text{with} \quad \mathbf{t}_X = (t_{X,1}, \dots, t_{X,n}) \in \mathbb{Z}_p^n, \\ &\quad \mathbf{t}_Z = (t_{Z,1}, \dots, t_{Z,n}) \in \mathbb{Z}_p^n. \end{aligned}$$

With this convention, the symplectic inner product of  $\mathbf{t}$  and  $\mathbf{t}'$  can be written as

$$\langle \mathbf{t}, \mathbf{t}' \rangle \triangleq \begin{bmatrix} t_{X,1} \\ t_{Z,1} \\ \vdots \\ t_{X,n} \\ t_{Z,n} \end{bmatrix}^T \cdot \begin{bmatrix} 0 & 1 \\ 1 & 0 & & \\ & & \ddots & \\ & & & 0 & 1 \\ & & & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} t'_{X,1} \\ t'_{Z,1} \\ \vdots \\ t'_{X,n} \\ t'_{Z,n} \end{bmatrix}$$

or as

$$\begin{aligned} \langle \mathbf{t}, \mathbf{t}' \rangle &= \begin{bmatrix} \mathbf{t}_X^T \\ \mathbf{t}_Z^T \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} & \mathbb{1}_n \\ \mathbb{1}_n & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} (\mathbf{t}'_X)^T \\ (\mathbf{t}'_Z)^T \end{bmatrix} \\ &= \mathbf{t}_X \cdot (\mathbf{t}'_Z)^T + \mathbf{t}_Z \cdot (\mathbf{t}'_X)^T, \end{aligned}$$

where  $\mathbb{1}_n$  is the  $n \times n$  identity matrix. Similar expressions will also be used for the vector  $\mathbf{m}$  and combinations of  $\mathbf{t}$  and  $\mathbf{m}$ .

**Definition 2** The dual code  $\mathcal{C}^\perp$  (under the symplectic inner product) of some group code  $\mathcal{C} \subseteq \prod_{i=1}^T \mathcal{T}_i$  is defined to be

$$\mathcal{C}^\perp \triangleq \left\{ \mathbf{t}' \in \prod_{i=1}^T \mathcal{T}_i \mid \langle \mathbf{t}, \mathbf{t}' \rangle = 0 \right\}.$$

□

Note that  $\mathcal{C}^\perp$  is also a group code and that  $(\mathcal{C}^\perp)^\perp = \mathcal{C}$ . Similarly, for any  $i = 1, \dots, T$ , because  $\mathcal{C}_i$  was assumed to be a group code, we can define its dual  $\mathcal{C}_i^\perp$ .

In the following, we want to show that there is an FFG representing  $\mathcal{C}^\perp$  that is tightly related to the FFG that represents  $\mathcal{C}$ . Continuing our example from Section II, let  $\mathcal{M}_i'' \triangleq \mathcal{M}_i'$  for  $i = 1, \dots, M$ , and let  $f^\perp : \prod_{i=1}^T \mathcal{T}_i' \times \prod_{i=1}^M \mathcal{M}_i' \times \prod_{i=1}^M \mathcal{M}_i'' \rightarrow \mathbb{R}$  be the function that represents the mapping

$$\begin{aligned} (\mathbf{t}', \mathbf{m}', \mathbf{m}'') &\mapsto f_1^\perp(t'_1, m'_1, m''_3) f_2^\perp(t'_2, m'_1, m'_2) \\ &\quad \cdot f_3^\perp(t'_3, t'_4, m''_2, m'_3) \\ &\quad \cdot [m'_1 = -m''_1] [m'_2 = -m''_2] [m'_3 = -m''_3] \end{aligned}$$

with

$$\begin{aligned} f_1^\perp(t'_1, m'_1, m''_3) &\triangleq [(t'_1, m'_1, m''_3) \in \mathcal{C}_1^\perp], \\ f_2^\perp(t'_2, m'_1, m'_2) &\triangleq [(t'_2, m'_1, m'_2) \in \mathcal{C}_2^\perp], \\ f_3^\perp(t'_3, t'_4, m''_2, m'_3) &\triangleq [(t'_3, t'_4, m''_2, m'_3) \in \mathcal{C}_3^\perp]. \end{aligned}$$

The function  $f^\perp$  is depicted by the FFG in Figure 1 (middle) where the function nodes with a tilde in them represent the indicator functions  $[m'_1 = -m''_1]$ ,  $[m'_2 = -m''_2]$ , and  $[m'_3 = -m''_3]$ , respectively. With this, we can follow [8] and establish the next theorem.

**Theorem 3** With the above definitions,

$$\mathcal{C}^\perp = \left\{ \mathbf{t}' \in \prod_{i=1}^T \mathcal{T}_i' \mid \begin{array}{l} \text{there exists } \mathbf{m}' \text{ and } \mathbf{m}'' \\ \text{such that } f^\perp(\mathbf{t}', \mathbf{m}', \mathbf{m}'') = 1 \end{array} \right\}.$$

*Proof:* (The proof is for the example code in Figure 1, but the proof can easily be generalized.) Let  $\mathbf{t}' \in \prod_{i=1}^T \mathcal{T}_i'$  be

such that there exist  $\mathbf{m}'$  and  $\mathbf{m}''$  such that  $f^\perp(\mathbf{t}', \mathbf{m}', \mathbf{m}'') = 1$ . Moreover, let  $\mathbf{t} \in \mathcal{C}$  and let  $\mathbf{m}$  be such that  $f(\mathbf{t}, \mathbf{m}) = 1$ . Then,

$$\begin{aligned} \langle \mathbf{t}, \mathbf{t}' \rangle &= \sum_{i=1}^T \langle t_i, t'_i \rangle \\ &= \sum_{i=1}^T \langle t_i, t'_i \rangle + \sum_{i=1}^M \langle m_i, m'_i \rangle - \sum_{i=1}^M \langle m_i, m'_i \rangle \\ &= \sum_{i=1}^T \langle t_i, t'_i \rangle + \sum_{i=1}^M \langle m_i, m'_i \rangle + \sum_{i=1}^M \langle m_i, m''_i \rangle \\ &= (\langle t_1, t'_1 \rangle + \langle m_1, m'_1 \rangle + \langle m_3, m''_3 \rangle) + \\ &\quad (\langle t_2, t'_2 \rangle + \langle m_1, m'_1 \rangle + \langle m_2, m'_2 \rangle) + \\ &\quad (\langle t_3, t'_3 \rangle + \langle t_4, t'_4 \rangle + \langle m_2, m''_2 \rangle + \langle m_3, m'_3 \rangle) \\ &\stackrel{(*)}{=} 0 + 0 + 0 = 0. \end{aligned}$$

Here, step (\*) follows from the fact that  $(t_1, m_1, m_3) \in \mathcal{C}_1$  and  $(t'_1, m'_1, m''_3) \in \mathcal{C}_1^\perp$  imply that  $\langle t_1, t'_1 \rangle + \langle m_1, m'_1 \rangle + \langle m_3, m''_3 \rangle = 0$ , with similar expressions for the other subcodes.

We see that  $\mathbf{t}'$  is orthogonal to  $\mathbf{t}$ , and because  $\mathbf{t} \in \mathcal{C}$  was arbitrary,  $\mathbf{t}'$  must be in  $\mathcal{C}^\perp$ . ■

**Assumption 4** For the rest of the paper we will assume that  $p = 2$ , which implies that the groups  $\mathcal{T}_i, \mathcal{T}_i', i = 1, \dots, T$ , and the groups  $\mathcal{M}_i, \mathcal{M}_i', \mathcal{M}_i'', i = 1, \dots, M$ , have characteristic 2. Therefore,  $m''_i = -m''_i$  for all  $m''_i \in \mathcal{M}_i'', i = 1, \dots, M$ , etc..

So, given that we are only interested in arguments of  $f^\perp$  that lead to non-zero function values, any valid configuration  $(\mathbf{t}', \mathbf{m}', \mathbf{m}'')$  of the FFG in Figure 1 (middle) fulfills  $\mathbf{m}' = -\mathbf{m}'' = \mathbf{m}''$ . This observation allows us to simplify the function  $f^\perp$  to  $f^\perp : \prod_{i=1}^T \mathcal{T}_i' \times \prod_{i=1}^M \mathcal{M}_i' \rightarrow \mathbb{R}$  that represents the mapping

$$(\mathbf{t}', \mathbf{m}') \mapsto f_1^\perp(t'_1, m'_1, m'_3) f_2^\perp(t'_2, m'_1, m'_2) f_3^\perp(t'_3, t'_4, m'_2, m'_3)$$

with

$$\begin{aligned} f_1^\perp(t'_1, m'_1, m'_3) &\triangleq [(t'_1, m'_1, m'_3) \in \mathcal{C}_1^\perp], \\ f_2^\perp(t'_2, m'_1, m'_2) &\triangleq [(t'_2, m'_1, m'_2) \in \mathcal{C}_2^\perp], \\ f_3^\perp(t'_3, t'_4, m'_2, m'_3) &\triangleq [(t'_3, t'_4, m'_2, m'_3) \in \mathcal{C}_3^\perp]. \end{aligned}$$

The new function  $f^\perp$  is depicted by the FFG in Figure 1 (right). It is clear that Theorem 3 simplifies to the following corollary.

**Corollary 5** With the above definitions and Assumption 4 we have

$$\mathcal{C}^\perp = \left\{ \mathbf{t}' \in \prod_{i=1}^T \mathcal{T}_i' \mid \begin{array}{l} \text{there exists an } \mathbf{m}' \\ \text{such that } f^\perp(\mathbf{t}', \mathbf{m}') = 1 \end{array} \right\}.$$

*Proof:* Follows easily from Theorem 3. ■

We conclude this section with a definition that will be crucial for the remainder of this paper, namely self-orthogonality and self-duality (under the symplectic inner product) of a code.

**Definition 6** Let  $\mathcal{C}$  be a group code with dual code  $\mathcal{C}^\perp$ . Then,

- $\mathcal{C}$  is called *self-orthogonal* if  $\mathcal{C} \subseteq \mathcal{C}^\perp$  and
- $\mathcal{C}$  is called *self-dual* if  $\mathcal{C} = \mathcal{C}^\perp$ .

(Note that a code  $\mathcal{C}$  is self-orthogonal if  $\langle \mathbf{t}, \mathbf{t}' \rangle = 0$  for all  $\mathbf{t}, \mathbf{t}' \in \mathcal{C}$ .)  $\square$

#### IV. STABILIZER LABEL CODES AND NORMALIZER LABEL CODES

Let  $\mathcal{C}$  be a code over  $\mathbb{Z}_2^2$  that is self-orthogonal under the symplectic inner product. Without going into the details of the stabilizer QECC framework, such a code  $\mathcal{C}$  can be used to construct a stabilizer QECC. In that context, the codes  $\mathcal{C}$  and  $\mathcal{C}^\perp$  are called, respectively, the stabilizer label code and the normalizer label code associated with that stabilizer QECC.

**Proposition 7** Using the notation that has been introduced so far, in particular Definition 1 and Assumption 4, let  $\mathcal{C} \subseteq \prod_{i=1}^T \mathcal{T}_i$  be a group code whose indicator function is defined by an FFG containing half-edges  $T_i$ ,  $i = 1, \dots, T$ , full edges  $M_i$ ,  $i = 1, \dots, M$ , and function nodes  $f_i$ ,  $i = 1, \dots, F$ , where the latter are indicator functions of group codes  $\mathcal{C}_i$ ,  $i = 1, \dots, F$ . Then,

- $\mathcal{C}$  is self-orthogonal if all  $\mathcal{C}_i$  are self-orthogonal, and
- $\mathcal{C}$  is self-dual if all  $\mathcal{C}_i$  are self-dual.

*Proof:* First we consider the case where all  $\mathcal{C}_i$  are self-orthogonal. The code  $\mathcal{C}$  can be represented by an FFG like the FFG in Figure 1 (left). Let  $\mathbf{t}$  be a codeword in  $\mathcal{C}$  and let  $\mathbf{m}$  be such that  $f(\mathbf{t}, \mathbf{m}) = 1$ . Because of Definition 1, Assumption 4, and Corollary 5, its dual code  $\mathcal{C}^\perp$  can be represented by an FFG like the FFG in Figure 1 (right). Then, because the FFG in Figure 1 (left) is topologically equivalent to the FFG in Figure 1 (right) and because all  $\mathcal{C}_i$  are self-orthogonal, it follows that  $f^\perp(\mathbf{t}, \mathbf{m}) = 1$ , which in turn yields  $\mathbf{t} \in \mathcal{C}^\perp$ . Finally, because  $\mathbf{t} \in \mathcal{C}$  was arbitrary, we see that  $\mathcal{C} \subseteq \mathcal{C}^\perp$ , i.e., that  $\mathcal{C}$  is self-orthogonal.

Secondly, we consider the case where all  $\mathcal{C}_i$  are self-dual. Similarly to the above argument, we can show that  $\mathcal{C} \subseteq \mathcal{C}^\perp$ . Reversing the roles of  $\mathcal{C}$  and  $\mathcal{C}^\perp$ , we can also show that  $\mathcal{C}^\perp \subseteq \mathcal{C}$ . This proves that  $\mathcal{C} = \mathcal{C}^\perp$ , i.e., that  $\mathcal{C}$  is self-dual.  $\blacksquare$

Obviously, Proposition 7 gives us a simple tool to construct stabilizer label codes and normalizer label codes. It does not seem that duality results for FFGs, which are at the heart of Proposition 7, have been leveraged before to construct stabilizer QECCs.<sup>3</sup>

##### A. CSS Codes

CSS codes are a family of stabilizer QECCs named after Calderbank, Shor, and Steane (see e.g. [2]). For these codes we will not use our formalism, however, later on CSS codes can be used as component codes for longer codes.

<sup>3</sup>While preparing this paper, we became aware of the recent paper [10] which also uses factor graphs and some type of duality results in the context of stabilizer QECCs. However, that paper does not give enough details for one to be able to judge its merits towards constructing stabilizer QECCs.

Let  $\mathcal{B}_1 \subseteq \mathbb{Z}_2^n$  and  $\mathcal{B}_2 \subseteq \mathbb{Z}_2^n$  be two binary codes of length  $n$  such that  $\mathbf{b} \cdot (\mathbf{b}')^T = 0$  for all  $\mathbf{b} \in \mathcal{B}_1$  and  $\mathbf{b}' \in \mathcal{B}_2$ . Based on these two binary codes, we define the stabilizer label code

$$\mathcal{C} \triangleq \{\mathbf{t} = (\mathbf{t}_X, \mathbf{t}_Z) \mid \mathbf{t}_X \in \mathcal{B}_1 \text{ and } \mathbf{t}_Z \in \mathcal{B}_2\}.$$

It can easily be seen that the code  $\mathcal{C}$  is self-orthogonal. Namely, for any  $\mathbf{t} = (\mathbf{t}_X, \mathbf{t}_Z)$ ,  $\mathbf{t}' = (\mathbf{t}'_X, \mathbf{t}'_Z) \in \mathcal{C}$  we have  $\langle \mathbf{t}, \mathbf{t}' \rangle = \mathbf{t}_X \cdot (\mathbf{t}'_Z)^T + \mathbf{t}_Z \cdot (\mathbf{t}'_X)^T = 0 + 0 = 0$ , where  $\mathbf{t}_X \cdot (\mathbf{t}'_Z)^T = 0$  follows from  $\mathbf{t}_X \in \mathcal{B}_1$  and  $\mathbf{t}'_Z \in \mathcal{B}_2$ , and where  $\mathbf{t}_Z \cdot (\mathbf{t}'_X)^T = \mathbf{t}'_X \cdot (\mathbf{t}_Z)^T = 0$  follows from  $\mathbf{t}'_X \in \mathcal{B}_1$  and  $\mathbf{t}_Z \in \mathcal{B}_2$ .

**Example 8** The so-called seven qubit Steane stabilizer QECC (see e.g. [2]) is a CSS code where both  $\mathcal{B}_1$  and  $\mathcal{B}_2$  equal the  $[7, 3, 4]$  binary simplex code, i.e., the code given by the rowspan of the matrix

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

$\square$

##### B. Codes over $\mathbb{F}_4$

Let us associate with any vector  $\mathbf{t} = (\mathbf{t}_X, \mathbf{t}_Z) \in (\mathbb{Z}_2^n)^2$  the vector  $\mathbf{t}_{\mathbb{F}_4} = (t_{\mathbb{F}_4,1}, \dots, t_{\mathbb{F}_4,n}) \in \mathbb{F}_4^n$  through the mapping<sup>4</sup>

$$\mathbf{t}_{\mathbb{F}_4} \triangleq \gamma_{\mathbb{Z}_2^2 \rightarrow \mathbb{F}_4}(\mathbf{t}) \triangleq \omega \mathbf{t}_X + \overline{\omega} \mathbf{t}_Z.$$

Clearly, the mapping  $\gamma_{\mathbb{Z}_2^2 \rightarrow \mathbb{F}_4}$  is injective and surjective and therefore bijective, and so there is a bijective mapping between  $\mathcal{C}$  and  $\mathcal{C}_{\mathbb{F}_4} \triangleq \gamma_{\mathbb{Z}_2^2 \rightarrow \mathbb{F}_4}(\mathcal{C})$ , the latter being the image of  $\mathcal{C}$  under the mapping  $\gamma_{\mathbb{Z}_2^2 \rightarrow \mathbb{F}_4}$ . Because  $\mathcal{C}$  was assumed to be a group/additive code,  $\mathcal{C}_{\mathbb{F}_4}$  is also a group/additive code. Moreover, if for any codeword  $\mathbf{t}_{\mathbb{F}_4} \in \mathcal{C}_{\mathbb{F}_4}$  it holds that  $\omega \cdot \mathbf{t}_{\mathbb{F}_4} \in \mathcal{C}_{\mathbb{F}_4}$ , then the code  $\mathcal{C}_{\mathbb{F}_4}$  is a linear code, i.e., not only is the sum of two codewords again a codeword, but any  $\mathbb{F}_4$ -multiple of a codeword is also a codeword. If  $\mathcal{C}_{\mathbb{F}_4}$  is a linear code then also  $\mathcal{C}_{\mathbb{F}_4}^\perp$  is a linear code. With this, Proposition 7 can be suitably reformulated for sub-codes  $\mathcal{C}_{\mathbb{F}_4,i} \triangleq \gamma_{\mathbb{Z}_2^2 \rightarrow \mathbb{F}_4}(\mathcal{C}_i)$  that are linear codes over  $\mathbb{F}_4$ , whereby one can show that the symplectic inner product can be replaced by the Hermitian inner product [4]; we leave the details to the reader. Note that a necessary condition for the code  $\mathcal{C}_{\mathbb{F}_4}$  to be linear is that  $|\mathcal{C}_{\mathbb{F}_4}|$  is a power of 4, i.e., that also  $|\mathcal{C}|$  is a power of 4.

**Example 9** The so-called five qubit stabilizer QECC (see e.g. [2]) has a stabilizer label code  $\mathcal{C}_{\mathbb{F}_4}$  that is the  $\mathbb{Z}_2$ -rowspan of

$$\begin{bmatrix} \omega & \overline{\omega} & \overline{\omega} & \omega & 0 \\ 0 & \omega & \overline{\omega} & \overline{\omega} & \omega \\ \omega & 0 & \omega & \overline{\omega} & \overline{\omega} \\ \overline{\omega} & \omega & 0 & \omega & \overline{\omega} \end{bmatrix},$$

It can easily be checked that  $\mathcal{C}_{\mathbb{F}_4}$  is a linear code, which allows one to represent it as the  $\mathbb{F}_4$ -rowspan of

$$\begin{bmatrix} \omega & \overline{\omega} & \overline{\omega} & \omega & 0 \\ 0 & \omega & \overline{\omega} & \overline{\omega} & \omega \end{bmatrix}.$$

$\square$

<sup>4</sup>The definition of  $\mathbb{F}_4$  was given at the end of Section I.



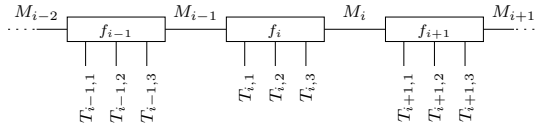


Fig. 2. FFG for the convolutional stabilizer label codes in Exs. 10 and 11.

## V. EXAMPLES

In this section we show how Proposition 7 can be leveraged to construct stabilizer label codes, in particular how that proposition unifies several earlier proposed stabilizer label code constructions. (For more details about the discussed codes we refer to the corresponding papers.)

**Example 10 (Convolutional Stab. QECC [6, Example 1])** With the help of our FFG framework, the stabilizer label code of [6, Example 1] can be seen to be given by the FFG in Figure 2, where, using the notation from Definition 1,  $\mu_i = 1$  for all  $i$ . For all  $i$ , the local function  $f_i$  is given by

$$f_i(m_{i-1}, t_{i,1}, t_{i,2}, t_{i,3}, m_i) \triangleq [(m_{i-1}, t_{i,1}, t_{i,2}, t_{i,3}, m_i) \in \mathcal{C}_i]$$

with  $\mathcal{C}_i$  such that  $\mathcal{C}_{\mathbb{F}_4, i} \triangleq \gamma_{\mathbb{Z}_2^2 \rightarrow \mathbb{F}_4}(\mathcal{C}_i)$  is a linear code that is the  $\mathbb{F}_4$ -rowspan of the matrix

$$\left[ \begin{array}{c|ccc|c} 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & \omega & \bar{\omega} & 0 \end{array} \right].$$

(In order to obtain a block code one needs to terminate the FFG in Figure 2 on both sides; we omit the discussion of this issue. Alternatively, tail-biting can be used.)  $\square$

**Example 11 (Convolutional Stab. QECC [6, Example 3])** Similarly, we can represent the stabilizer label code of [6, Example 3] by the FFG in Figure 2. Here, however, we have  $\mu_i = 2$  for all  $i$ , and  $\mathcal{C}_i$  is such that  $\mathcal{C}_{\mathbb{F}_4, i} \triangleq \gamma_{\mathbb{Z}_2^2 \rightarrow \mathbb{F}_4}(\mathcal{C}_i)$  is a linear code given by the  $\mathbb{F}_4$ -rowspan of

$$\left[ \begin{array}{cc|ccc|cc} 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{array} \right].$$

Note that a “more common” choice for a matrix whose  $\mathbb{F}_4$ -rowspan is the trellis section code of a non-recursive convolutional code would have been a matrix like

$$\left[ \begin{array}{cc|ccc|cc} 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \end{array} \right].$$

Here the rows are such that the last  $\mu_i - 1$  components of  $m_{i-1}$  equal the first  $\mu_i - 1$  components of  $m_i$ . However, the  $\mathbb{F}_4$ -span of such a matrix does not result in a self-orthogonal  $\mathcal{C}_i$ .  $\square$

**Example 12 (Serial Turbo Stab. QECC [7, Figure 10])** The paper [7] discusses constructions of serial turbo stabilizer label codes. In particular, Figure 10 in [7] presents a code that corresponds to the FFG shown in Figure 3 (left). Here,  $f_{\text{convcode1}}$ ,  $f_{\text{quantum-interleaver}}$ , and  $f_{\text{convcode2}}$  represent, respectively, the

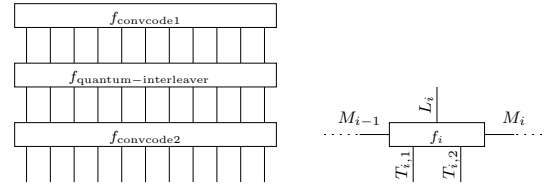


Fig. 3. Left: FFG for the serial turbo stabilizer label code in Example 12. Right: FFG for the second convolutional code on the left-hand side.

indicator functions of the first convolutional code, of the quantum interleaver, and of the second convolutional code. If the indicator functions correspond to self-orthogonal codes then we can apply our FFG framework and guarantee that the overall code is self-orthogonal. This is indeed the case for the codes presented in [7].

A particular example of a stabilizer label code that can be used for the function node  $f_{\text{convcode2}}$  is given in [7, Figures 8 and 9] and shown as an FFG in Figure 3 (right). (In contrast to [7, Figures 8 and 9], that uses the variable names  $P_{i,1}$  and  $P_{i,2}$ , we are using the variable names  $T_{i,1}$  and  $T_{i,2}$ , respectively.) Here, for all  $i$  the indicator function  $f_i$  corresponds to a code  $\mathcal{C}_i$  that is the rowspan of

$$\left[ \begin{array}{cccc|ccccc} 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array} \right],$$

where the columns correspond to  $m_{X,i-1}$ ,  $l_{X,i}$ ,  $t_{X,i,1}$ ,  $t_{X,i,2}$ ,  $m_{X,i}$ ,  $m_{Z,i-1}$ ,  $l_{Z,i}$ ,  $t_{Z,i,1}$ ,  $t_{Z,i,2}$ ,  $m_{Z,i}$ , respectively. Note that  $\mathcal{C}_i$  is self-dual under the symplectic inner product and that the corresponding  $\mathbb{F}_4$ -code  $\mathcal{C}_{\mathbb{F}_4, i} \triangleq \gamma_{\mathbb{Z}_2^2 \rightarrow \mathbb{F}_4}(\mathcal{C}_i)$  is additive but not linear. (It cannot be linear since the number of codewords is 32, which is not a power of 4.)  $\square$

**Example 13 (Stabilizer State [11, Example 1])** Roughly speaking, a stabilizer state corresponds to a stabilizer QECC whose stabilizer label code is self-dual, see [12], [13], [11], [14]. Let  $\mathbf{A}$  be the  $n \times n$  adjacency matrix of any graph with  $n$  vertices and let  $\mathcal{C}$  be the rowspan of  $[\mathbb{1}_n | \mathbf{A}]$ , where the columns correspond to  $t_{X,1}, \dots, t_{X,n}$ ,  $t_{Z,1}, \dots, t_{Z,n}$ , and where  $\mathbb{1}_n$  is the  $n \times n$  identity matrix. It can easily be checked that  $\mathcal{C}$  is self-dual under the symplectic inner product. (Note that  $\mathbf{A}^T = \mathbf{A}$  because  $\mathbf{A}$  is the adjacency matrix of a graph.)

For example, the graph in Figure 4 (left) results in a stabilizer label code  $\mathcal{C}$  which is the rowspan of

$$\left[ \begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{array} \right].$$

It turns out that any such stabilizer label code can also be represented by an FFG that is topographically closely related to the graph that defined the code. E.g., Figure 4 (right)

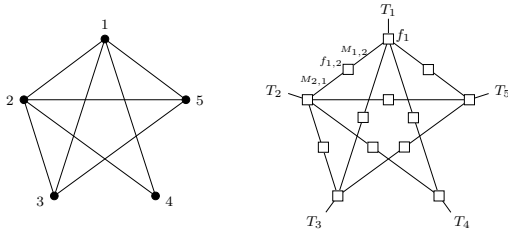


Fig. 4. Left: graph with five vertices that defines a stabilizer state. Right: FFG of the corresponding stabilizer label code.

shows the FFG that corresponds to the example graph in Figure 4 (left). Here,  $f_1$  is the indicator function of the self-dual code  $\mathcal{C}_1$  which is defined to be the rowspan of

$$\left[ \begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{array} \right],$$

where the columns correspond to  $t_{X,1}$ ,  $m_{X,1,2}$ ,  $m_{X,1,3}$ ,  $m_{X,1,4}$ ,  $m_{X,1,5}$ ,  $t_{Z,1}$ ,  $m_{Z,1,2}$ ,  $m_{Z,1,3}$ ,  $m_{Z,1,4}$ ,  $m_{Z,1,5}$ , respectively. Moreover,  $f_{1,2}$  is the indicator function of the self-dual code  $\mathcal{C}_{1,2}$  which is defined to be the rowspan of  $\left[ \begin{array}{cc|cc} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{array} \right]$ , where the columns correspond to  $m_{X,1,2}$ ,  $m_{X,2,1}$ ,  $m_{Z,1,2}$ ,  $m_{Z,2,1}$ , respectively. The other indicator functions  $f_i$  and  $f_{i,j}$  and self-dual codes  $\mathcal{C}_i$  and  $\mathcal{C}_{i,j}$  are defined analogously. Note that the code  $\mathcal{C}_i$  depends on the number of vertices that are adjacent to vertex  $i$ . However, the code  $\mathcal{C}_{i,j}$  is always the same for any pair  $(i, j)$  of adjacent vertices.  $\square$

**Example 14 (LDPC codes)** Any LDPC code whose parity-check matrix contains orthogonal rows can be used to construct a normalizer label code  $\mathcal{C}^\perp$ , see e.g. [5], [15], [16] and references therein. In terms of FFGs, such LDPC codes are expressed with the help of equal and single-parity-check function nodes. The FFG of the corresponding stabilizer label code  $\mathcal{C}$  can also be expressed in terms of equal and single-parity-check function nodes. Because single-parity-checks of length not equal to 2 do *not* represent self-orthogonal codes, our FFG framework is not directly applicable to construct FFGs of such LDPC codes. However, with the help of some auxiliary code constructions, our framework can also be used to construct LDPC stabilizer/normalizer label codes; because of space constraints we do not give the details here.  $\square$

We leave it as an open problem to use our FFG framework to construct other classes of stabilizer label codes that have interesting properties.

## VI. MESSAGE-PASSING ITERATIVE AND LINEAR PROGRAMMING DECODING

One of the main interests in studying FFGs for stabilizer label codes and their duals is that one would like to have FFGs that are suitable for MPI decoding. (Note that the code

that is relevant for decoding in the stabilizer QECC framework is the code  $\mathcal{C}^\perp$ , or the coset code  $\mathcal{C}^\perp/\mathcal{C}$ , see e.g. the comments in [7].) The well-known trade-offs from classical LDPC and turbo codes apply also here: good codes with low FFG variable and function node complexity must have cycles, yet cycles lead to sub-optimal performance of message-passing iterative decoders. We leave it as an open problem to study stopping sets, trapping sets, absorbing sets, near-codewords, pseudo-codewords, the fundamental polytope, etc. (see e.g. the refs. at [17]) for the codes that were discussed in this paper. Moreover, one can formulate alternative decoders to MPI decoders like linear programming (LP) decoding. It would be interesting to see if the self-orthogonality property of stabilizer label codes leads to further insights in the context of MPI and LP decoders, in particular by also leveraging other duality results for FFGs like Fourier duality [18] and Lagrange duality [19].

## REFERENCES

- [1] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. on Inform. Theory*, vol. IT-47, no. 2, pp. 498–519, Feb. 2001.
- [2] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, UK: Cambridge University Press, 2000.
- [3] D. Gottesman, "Stabilizer codes and quantum error correction," Ph.D. dissertation, California Institute of Technology, Pasadena, CA, USA, 1997.
- [4] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, "Quantum error correction via codes over  $\text{GF}(4)$ ," *IEEE Trans. on Inform. Theory*, vol. IT-44, no. 4, pp. 1369–1387, July 1998.
- [5] D. J. C. MacKay, G. Mitchison, and P. L. McFadden, "Sparse-graph codes for quantum error correction," *IEEE Trans. on Inform. Theory*, vol. IT-50, no. 10, pp. 2315–2330, Oct. 2004.
- [6] G. D. Forney, Jr., M. Grassl, and S. Guha, "Convolutional and tail-biting quantum error-correcting codes," *IEEE Trans. on Inform. Theory*, vol. IT-53, no. 3, pp. 865–880, Mar. 2007.
- [7] D. Poulin, J.-P. Tillich, and H. Ollivier, "Quantum serial turbo-codes," *submitted, available online under* <http://arxiv.org/abs/0712.2888>, Dec. 2007.
- [8] G. D. Forney, Jr., "Codes on graphs: normal realizations," *IEEE Trans. on Inform. Theory*, vol. IT-47, no. 2, pp. 520–548, Feb. 2001.
- [9] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Sig. Proc. Mag.*, vol. 21, no. 1, pp. 28–41, Jan. 2004.
- [10] H. Wang, J. Wang, Q. Du, and G. Zeng, "A new approach to constructing CSS codes based on factor graphs," *Information Sciences*, vol. 178, no. 7, pp. 1893–1902, Apr. 2008.
- [11] M. Van den Nest, J. Dehaene, and B. De Moor, "Graphical description of the action of local Clifford transformations on graph states," *Phys. Rev. A*, vol. 69, no. 2, pp. 022316.1–022316.7, 2004.
- [12] D. Schlingemann and R. F. Werner, "Quantum error-correcting codes associated with graphs," *Phys. Rev. A*, vol. 65, p. 012308, Dec. 2001.
- [13] D. Schlingemann, "Stabilizer codes can be realized as graph codes," *Quant. Inf. Comp.*, vol. 2, no. 4, pp. 307–323, June 2001.
- [14] A. Cross, G. Smith, J. A. Smolin, and B. Zeng, "Code-word stabilized quantum codes," *submitted, available online under* <http://arxiv.org/abs/0708.1021v1>, Aug. 2007.
- [15] S. A. Aly, "A class of quantum LDPC codes constructed from finite geometries," *submitted, available online under* <http://aps.arxiv.org/abs/0712.4115>, Dec. 2007.
- [16] I. Djordjevic, "Quantum LDPC codes from balanced incomplete block designs," *IEEE Comm. Letters*, vol. 12, no. 5, pp. 389–391, May 2008.
- [17] <http://www.pseudocodewords.info>.
- [18] Y. Mao and F. R. Kschischang, "On factor graphs and the Fourier transform," *IEEE Trans. on Inform. Theory*, vol. IT-51, no. 5, pp. 1635–1649, 2005.
- [19] P. O. Vontobel and H.-A. Loeliger, "On factor graphs and electrical networks," in *Mathematical Systems Theory in Biology, Communication, Computation, and Finance, IMA Volumes in Math. & Appl.*, D. Gilliam and J. Rosenthal, Eds. Springer Verlag, 2003.